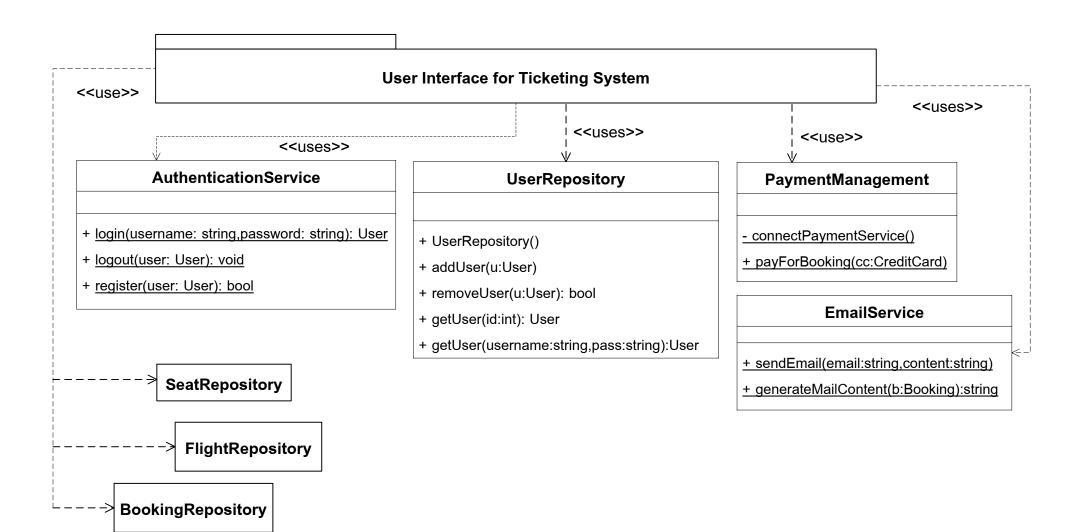
เดิม (ระหว่าง UI กับ BL)

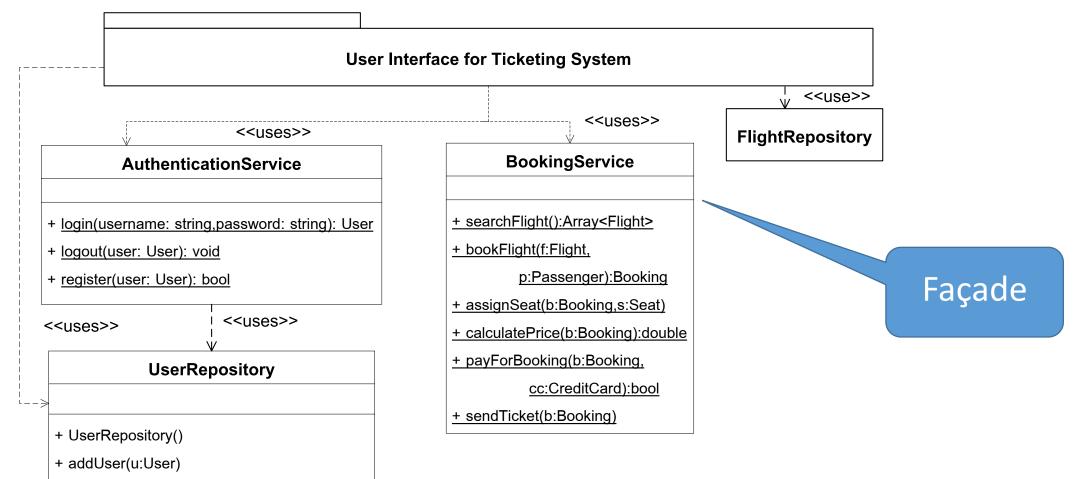


1. สร้าง Façade สำหรับการจองให้ UI ใช้ง่ายขึ้น

+ removeUser(u:User): bool

+ getUser(username:string,pass:string):User

+ getUser(id:int): User

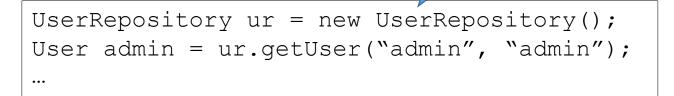


2. Repository ทุกตัว ทำให้เป็น Singleton ได้

์ อาจสร้างและทำลาย ในทุกครั้งที่ใช้

UserRepository

- + UserRepository()
- + addUser(u:User)
- + removeUser(u:User): bool
- + getUser(id:int): User
- + getUser(username:string,pass:string):User





UserRepository

- theInstance: UserRepository
- UserRepository()
- + getInstance: UserRepository
- + addUser(u:User)
- + removeUser(u:User): bool
- + getUser(id:int): User
- + getUser(username:string,pass:string):User

```
UserRepository ur = UserRepository.getInstance();
User admin = ur.getUser("admin", "admin");
...
```

3. แบ่ง Flight ออกเป็น 2 คลาส ด้วย Abstraction-Occurence

Flight

- flightNumber: string

- source: string

- destination: string

- departureTime: Time

- arrivalTime: Time

- date: Date

- price: double

+ Flight(fn: string, source: string ...)

+ setDepartureTime(time: Time)

+ setArrivalTime(time: Time)

+ setPrice(price: double)



RegularFlight

- flightNumber: string

- source: string

- destination: string

- departureTime: Time

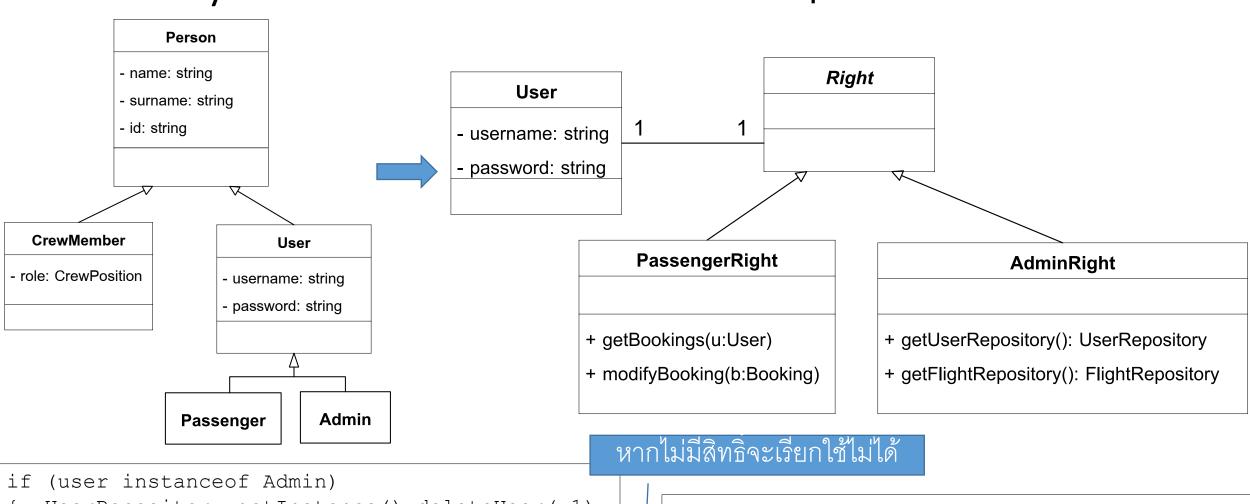
- arrivalTime: Time

- price: double

SpecificFlight

- date: Date

4. Player-Role สำหรับ User แบบต่างๆ



```
{ UserRepository.getInstance().deleteUser(u1);
    ...
}
```

```
AdminRight r = (AdminRight) user.getRight();
`r.getUserRepository().deleteUser(u1);
```

ส่งผลให้ UI เรียกใช้ Repository ต่างๆ โดยอ้อม แต่เรียกผ่าน User แทน (จาก AuthenticationService)

